

SZAKDOLGOZAT

Varga Marcell

2014

Pannon Egyetem
Matematika Tanszék
Mérnök informatikus BSc szak

SZAKDOLGOZAT

Képfeldolgozást támogató keretrendszer és
modulok készítése

Félkövr
for-
má-
zá-
sok

Varga Marcell

Témavezető: Lipovits Ágnes

2014

Ide jön az eredeti vagy a fénymásolt feladatkiírás.

Nyilatkozat

Alulírott Varga Marcell diplomázó hallgató kijelentem, hogy a szakdolgozatot a Pannon Egyetem Matematika Tanszékén készítettem Mérnök informatikus BSc szak (BSc in Computer Engineering) megszerzése érdekében.

Kijelentem, hogy a szakdolgozatban lévő érdemi rész saját munkám eredménye, az érdemi részen kívül csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy a szakdolgozatban foglalt eredményeket a Pannon Egyetem, valamint a feladatot kiíró szervezeti egység saját céljaira szabadon felhasználhatja.

Veszprém, 2014. május 02.

Aláírás

Alulírott Lipovits Ágnes témavezető kijelentem, hogy a szakdolgozatot Varga Marcell a Pannon Egyetem Matematika Tanszékén készítette Mérnök informatikus BSc szak (BSc in Computer Engineering) megszerzése érdekében.

Kijelentem, hogy a szakdolgozat védeésre bocsátását engedélyezem.

Veszprém, 2014. május 02.

Aláírás

Köszönetnyilvánítás

Köszönet!

TARTALMI ÖSSZEFOGLALÓ

E szakdolgozat témája ...

Kulcsszavak: szoftverarchitektúra, képfeldolgozás, adatszerkezetek, adatkezelés, Qt, c++, OpenCV

ABSTRACT

The topic of this thesis is to ...

Keywords: software-architecture, image processing, data structure, data processing, Qt, c++, OpenCV

Todo list

Félkövér formázások	2
usecase	8
Figure: INPUT USECASE ELEMENTS	8
Törölni!	9
Törölni!	9
Törölni!	9
Törölni!	9
USECASE	10
Licencelési infók	13

Tartalomjegyzék

1.	A feladat összefoglalása	1
1.1.	Első lépés	1
2.	Hasonló célú rendszerek	2
2.1.	Összehasonlítási szempontok	2
2.1.1.	Általános tulajdonságok:	2
2.1.2.	Képfeldolgozási képességek	4
2.2.	Választott szoftverek	4
2.3.	Összefoglalás	5
2.3.1.	Tapasztalatok	5
2.3.2.	Célok	6
3.	Rendszertervek	6
3.1.	Szószedet	7
3.2.	Modulok	8
3.2.1.	Input - Bemenet	8
3.2.2.	Plugin - Bővítmények	9
3.2.3.	Process Chain - Feldolgozás	9
3.2.4.	IMP -	9
3.3.	Követelmény analízis	9
3.3.1.	Funkcionális követelmények	9
3.3.2.	Nem funkcionális követelmények	10
3.4.	Választott technológiák	10
3.4.1.	Qt	11
3.4.2.	OpenCV	11
3.5.	Architektúra	12
3.6.	Felület terv	12

4.	Irodalomjegyzék	12
5.	Mellékletek	13
5.1.	Hasonló célú rendszerek összehasonlítása táblázat	14
5.2.	Teljes usecase	15
5.3.	CD melléklet	15

1. A FELADAT ÖSSZEFOGLALÁSA

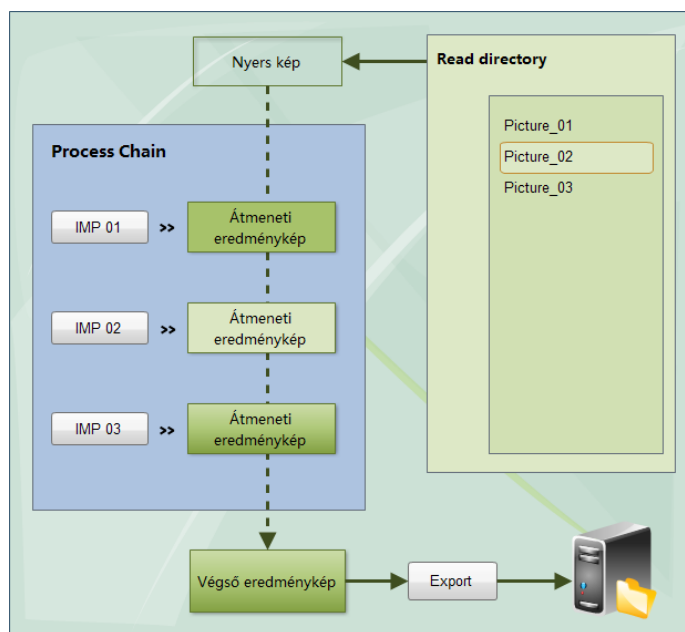
1. A feladat összefoglalása

Témám egy olyan képfeldolgozást támogató keretrendszer tervezése és fejlesztése, amely alkalmas képek egyedi vizsgálatára és kötegetelt feldolgozására. A feldolgozást végző algoritmusok a dinamikusan betölthető modulokban foglalnak helyet. A rendszer fő hasznélvezője a Pannon Egyetem Képfeldolgozás Kutatólaboratóriuma lesz, de célom, hogy kellően általános rendszer jöjjön létre, amelyet bárki könnyen és egyszerűen használhat, illetve bővítheti saját modulokkal.

1.1. Első lépés

A legtöbb munka során előnyös, ha projekt lényegét megragadva röviden összefoglaljuk a főlefutási eset sikeres teljesülését. Erre jó példa a rövid (brief[1]) formátumú usecase.

”A felhasználó összeállítja a bemeneti képek, adatforrások listáját. Ezek után meghatározza a feldolgozás lépéseit. Végül megjelöli a kimeneti formát, majd elindítja a feldolgozást. A program egyesével beolvassa a feldolgozni kívánt képeket, majd az adott képen végrehajtja sorrendhelyesen a kijelölt feladatokat, végül a választott kimeneti formába menti ki az eredményképeket.”



1. ábra. A rendszer vázlatos működése

2. HASONLÓ CÉLÚ RENDSZEREK

Az 1. ábrán láthatjuk a rendszer vázlatos működését, egy kezdeti terven. Jelen esetben egy könyvtárban található összes digitális képet kívánjuk feldolgozni. A beolvasás után a nyers képet¹ átadjuk a feldolgozást végző logikának (Process Chain), ahol jelen példában 3 darab elemi művelet történik (IMP01-03). Fontos megjegyezni, hogy a feldolgozás pipelining [2] jellegűt követ, tehát apró elemi lépések sorozata, amelyek kötött sorrendben végzik a feldolgozást. Természetesen a feldolgozás menete nem csak lineáris jellegű lehet hanem pl.: fa jellegű. (Ilyen műveleti módra jó példa különböző grafikus engineknél a fények számítása pl.: [3].) A feldolgozás utolsó lépése befejezését követően, a végső eredménykép jelen esetben exportálásra kerül egy a bementi könyvtárral nem megegyező könyvtárba.

2. Hasonló célú rendszerek

Következő lépésként megvizsgáltam, hogy milyen hasonló célú szoftverek, illetve szoftver csomagok találhatóak a piacon. Erre azért volt szükség, hogy pontosabb képet kapjak a jelenleg fellelhető megoldásokról, és munkám során az így tapasztalt pozitív és negatív tapasztalatokat felhasználva jó minőségű szoftvert fejleszthessek.

Különböző összehasonlítási szempontokat állítottam fel, melyek lentebb olvashatóak. Az vizsgálat során a személyes benyomáson túl, egyéni véleményeket is figyelembe vettem (pl.: kiadó cégnek vagy alapítványak az ajánlása, vagy független publikáció, újságcikk).

2.1. Összehasonlítási szempontok

2.1.1. Általános tulajdonságok:

- Platform: Milyen környezetben és operációs rendszeren használható? Milyen program nyelvvel fejlesztették?

Hordozhatósági szempontonból került be a listára. Oka, hogy sokkal

¹Nyersképnek nevezzük minden olyan bemeneti képet, amelyen nem hajtottunk végre semmiféle változtatást.

2. HASONLÓ CÉLÚ RENDSZEREK

könnyebb egy olyan rendszert használni, amely több környezetben is elboldogul.

- Licence: Milyen licenc alatt került publikálásra?
Elsősorban pénzügyi és kód újrahasznosíthatóság miatt érdekes.
- Cél csoport: A szoftver kinek az igényeinek teljesítésére törekszik?
Legtöbb esetben a célcsoport már alapvetően meghatározza, hogy a szoftverbe milyen funkcionalitásokat építünk be, illetve, hogy ezekhez milyen interfészt biztosítunk a jövőbeni felhasználók számára.
- Támogató: Van hivatalos támogatottságga? (cég, alapítvány)
Az esetek jelentős részében megfigyelhető, hogy egy szoftver, szoftver-csomag akkor válik igazán jól támogatottá, ha fejlesztői közösségen kívül egy nagyobb szervezet is gondozásába veszi.
- Felhasználói közösség: Fórum, levelező listák?
Bármilyen előre nem látható hiba történhet: Ami elromolhat az el is romlik! A fenti csatornákon segítséget kérve nagy eséllyel kaphatunk választ kérdésünkre, és megoldást problémáinkra.
- Plugin rendszer: Plugin betöltésre van lehetőségünk? Saját plugin?
A képfeldolgozás egy eléggé sokrétű szerteágazó lehetőségeket, funkcionalitásokat magában foglaló szakterület. Így az csak utópisztikus álom, hogy egyszer valaki implementálja az összes funkcionalitást és onnantól kezdve mindenki boldogan használja azokat az idő végezetéig... Ezért ha a program dinamikusan bővíthető (akár a felhasználó által készített bővítményekkel), jelentős előnyt jelent a többi monolitikus rendszerrel szemben.
- Kötegetelt feldolgozási lehetőség: Feldolgozhatunk egyszerre nagy mennyiségű képet?
Senki nem fog egyesével feldolgozni nem, hogy 50.000 darab képet de még 500-at sem...
- Automatizálási lehetőségek: Automatizálhatjuk a feldolgozást?
Ha lehetőségünk van a meglévő egyszerű feldolgozási lépéseket testreszabni, esetleg összekombinálni akkor az szintén egy jelentős előny lehet.

2. HASONLÓ CÉLÚ RENDSZEREK

- Fejlesztői eszközök: Rendelkezik hivatalos fejlesztői eszközökkel?
Köztudott, hogy amelyik szoftverhez kiadásra kerülnek különböző fejlesztői eszközök, és segédanyagok ahhoz lényegesen egyszerűbb saját modulokat implementálni.
- Támogatott bemeneti formátumok köre
- Megjelenítési, vizualizációs lehetőségek listája, módjai
Hasznos ha többféle interfészt biztosítunk az adott információ megjelenítéshez: más logikai kontextusban helyezve új megfigyeléseket is tehet a felhasználónk.

2.1.2. Képfeldolgozási képességek

- Képjavító eljárások, pl.: élesítés, kontrasztkiegyenlítés
- Geometriai műveletek, pl.: átméretezés, forgatás, tükrözés
- Analizálás, pl.: eltérések detektálása, alacsony szintű képleírók
- Szerkesztési műveletek, pl.: logikai, szöveg, alakzatok elhelyezése
- Színterek közötti konverzió, pl.: RGB \rightarrow HSL, csatornák külön kezelése stb

2.2. Választott szoftverek

- ImageJ[4]
Képfeldolgozást és analízálást végző rendszer, amely a National Institutes of Health fejlesztése. A program első indulásakor látható, hogy itt egy professzionális orvostechnológiai eszközről van szó. Támogatottsága jelentős mind közösségi, mind bővíthetőségi szempontból. Fegyvertárában olyan eszközöket is felvonultat mint Z és T funkciók.[5]
A Z funkciók segítségével pl.: MRI-vel készített sorozatos metszeti képeket kezelhetünk könnyedén, lehetőségeinket tovább növeli, hogy a térbeli szervezés mellett még időbeli struktúra felépítésére és kezelésére is lehetőséget ad a program (T funkciók).

2. HASONLÓ CÉLÚ RENDSZEREK

- ImBatch[6]

Képfeldolgozást végző rendszer. Célcsoportja egyértelműen egy félprofesszionális felhasználói szint. Tehát itt eleve nem is várunk professzionális analitikai funkciókat. Cserébe kapunk egy szép, letisztult, egyszerű grafikus felhasználói felületet, és egy pár használatot segítő kényelmi funkciót: pl.: Windows helyérzékeny menü integrációt.

- OriginLab - Image Processing[7]

Az OriginLab szoftver csomag része, amely első sorban tudományos és ipari célközöséget szolgál ki. [8] A korábban tárgyalt rendszerekkel ellentétben ez a szoftver fizetős (21 napos teszt verzió igényelhető). Ára hozza az iparban szokásos szoftver árakat [9], amely személyes felhasználásra kissé borsos, azonban funkcionalitása kárpótolja a felhasználót. Rentgegen elemzési lehetőség mellett még OriginC-ben saját algoritmusainkat is megvalósíthatunk, a LabView támogatás már majdnem, hogy csak hab a tortán.

2.3. Összefoglalás

2.3.1. Tapasztalatok

A részletes összehasonlítás az 1.2. táblázatból olvasható ki a 14. oldalon.

Az adatsorokból elemzése közben, több fontos követelmény, fejlesztési irányvonal körvonalazódott:

- Hordozható legyen több platformra. Hiszen egy laboratóriumban többféle architektúra elő fordul. (Ideális esetben tehát a szoftverünk legyen crossplatform.)
- Nyitott legyen a további fejlesztésekre. A felhasználónak adjuk meg a lehetőséget, hogy testre szabhassa a szoftverünket, vagy akár önmaga is fejlesztővé válhasson, így bővíthesse a pluginek körét, vagy javítsa a fő programot.
- Szerepeljenek automatizálási lehetőségek. Írhassunk makrókat, vagy vizuális módon szerkeszthessünk algoritmusokat.

3. RENDSZERTERVEK

- A rendszer ajánlott a célcsoport, munkája közben gyakorta előforduló bemeneti formátum kezelésére fel kell készíteni.
- Már gyárilag nagy mennyiségű képjavító, feldolgozó, szerkesztő, elemző és analízáló funkcionalitással érkezzen a szoftver.

2.3.2. Célok

Összegezve láthatjuk, hogy eléggé könnyen lehet már előkészített rendszereket választani a szoftverpiac palettájáról. Ilyenkor jogosan felmerül a kérdés, hogy ez a projekt miben ad többet, mint a jelenlegi lehetőségek?

- Célom, hogy egy egyszerű felhasználó számára is könnyen kezelhető nyitott bővíthető rendszer hozzak létre, ahol többszintű képfeldolgozást is egyszerűen végezhetjük.
- A programba új funkciók integrálása ne ütközzön problémákba, és az ilyen módon implementált funkcióknak a főprogramtól függetlenül is terjesztőknek kell lenniük. Ez természetesen maga után vonzza, hogy szükséges kezelni a későbbi továbbfejlesztések esetén elképzelhető plugin kompatibilitási problémákat.
- A nagy mennyiségű kisméretű funkció egy idő után sajnos kezelhetetlenné bizonyul, ezért azokat funkcionalitásuk alapján csoportosítani kell.
- Legyen lehetőség több különböző szerkezetű grafikus felhasználói felület használatára. Előzetes tervekben egy blokkos-kapcsolat alapú grafikus felület szerepel. Ez a módszer több különböző technológia területen sikeresen vizsgázott pl.: Labview blockdiagrammjai [11], vagy UDK4 Blueprint Editorja[12] (amely az UDK3 Kismetjének egy tovább fejlesztett változata).

3. Rendszertervek

A következőkben röviden összefoglalom a szakdolgozat tárgyát képező szoftver terveinek elő- és elkészítésének menetét. A korábbi fejezet végén vázolni

3. RENDSZERTERVEK

kezdtém a programmal szembeni elvárásokat, javaslatokat, ez gyakorlatilag a követelményrendszer felírásának a kezdeti lépése volt. Természetesen a jelenlegi követelményrendszer kialakítását még megelőzte több beszélgetés, konzultáció is.

A megbeszélések során több logikailag összetartozó objektum, folyamat került felírásra. Ezek rendre nevet kaptak a kommunikáció hatékonyságának növelése érdekében. A legfontosabb elnevezéseket a 1.1 táblázat prezentálja.

A továbbiakban ezeket az elnevezéseket használom az adott objektumokra.

3.1. Szószedet

Elnevezés	Definíció
Bimg	A főprogramnak az elnevezése (képzése a Batch Image szavakból történt szóösszerántással).
Modul	BIMG-n belüli logikai egység.
Plugin	BIMG dinamikus kiterjesztése, az IMP blockok lelőhelye
IMP Block	Image Process Block, képfeldolgozási alapegységnek tekinthetjük, egy IMP általában egy jól körülhatárolt művelet elvégzésére alkalmas
IMP Parameter	Megkülönböztetünk: bemeneti paramétereket (pl.: IMP bemeneti kép), kimeneti paramétereket (pl.: IMP eredménykép) és konfigurációs paramétereket (pl.: konstans, mátrix).
IMP Connection	Kettő darab azonos típusú IMP Parameter között kapcsolat hozható létre. Ezt a kapcsolatot nevezzük IMP Connection-nek. Kapcsolat csak kimeneti és bemeneti paraméterek között jöhet létre. Az irány kötelezően: Ki-Be.
IMP Node	Egy IMP Blockot tartalmaz. Feladata, hogy megfelelő interfészt biztosítson az IMP Block fölé.
IMP Connector	Egy IMP Parameter-t tartalmaz. Feladata, hogy megfelelő interfészt biztosítson az IMP Parameter fölé.
IMP Connector Connection	Egy IMP Connection-t tartalmaz. Feladata, hogy megfelelő interfészt biztosítson az IMP Connection fölé.
BimgImage	Be- és kimeneti adatsort reprezentál.
Process Chain	IMP-ket tartalmaz, itt történik az IMP-k meghívása feldolgozáskor

1.1. táblázat. Szószedet

3. RENDSZERTERVEK

3.2. Modulok

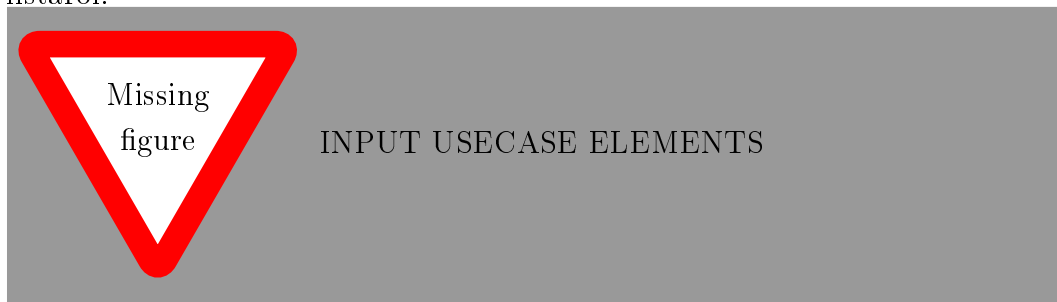
Funkcionális szempontok alapján több modul definiálása történt meg. A következőkben ezeket a modulokat vázolom. Részletesebb leírás a usecase diagramm-ról olvasható le. (Illetve a mellékletben csatolt usecase-ben.)

usecase

3.2.1. Input - Bemenet

Itt történik a bemeneti adatsorok összeválogatása, és rendezése.

”A felhasználó saját igényei szerint összeállíthat egy bemeneti listát, amely a feldolgozni kívánt képeket, adatsorokat tartalmazza. A listát bővítheti egy vagy több elemmel, továbbá megadhat pl.: mappát, amiben az összes megfelelő formátumú elemet hozzáfűzi rendszer a bemeneti listához. Amelyik elem már szerepel a listán az nem kerül újra hozzáfűzésre a listához. A felhasználónak lehetősége van továbbá egyszerre egy elem kiválasztására, ezt az elemet eltávolíthatja a listáról és módosíthatja a listában betöltött helyét. A felhasználónak lehetősége van kiüríteni az egész listát, továbbá el is mentheti a lista tartalmát egy a rendszeren kívüli erőforrásra, amelynek segítségével később a lista újra betölthető lesz. Betöltéskor a nem elérhető elemeket a rendszer eltávolítja a listáról.”



3. RENDSZERTERVEK

3.2.2. Plugin - Bővítmények

3.2.3. Process Chain - Feldolgozás

3.2.4. IMP -

3.3. Követelmény analízis

A követelmény analízist a RUP (Rational Unified Process) metodika FURPS+ rendszere alapján végeztem.

Így minden olyan tulajdonság és jellemző feltüntetésre került amely szükséges, hogy a szoftver megoldja az adott problémát. [19] Ezek a követelmények jellemzőjüket tekintve lehetnek funkcionális követelmények, és nem funkcionális követelmények.

3.3.1. Funkcionális követelmények

Funkcionális követelményeknek tekintünk minden olyan követelményt, amely a fő termékünkbe valamilyen képességet biztosít. pl.: Adott objektum megjelenítése [20]

A 2. ábrán megfigyelhető a sematikus usecase diagrammja. (A teljes diagramm a 3. ábrán és a 4. ábrán tekinthető meg a 16. oldalon. Az egy logikai csoportba tartozó funkciókat csoportokba szerveztem. Így a következő csoportok kerültek felírásra:

Törölni!

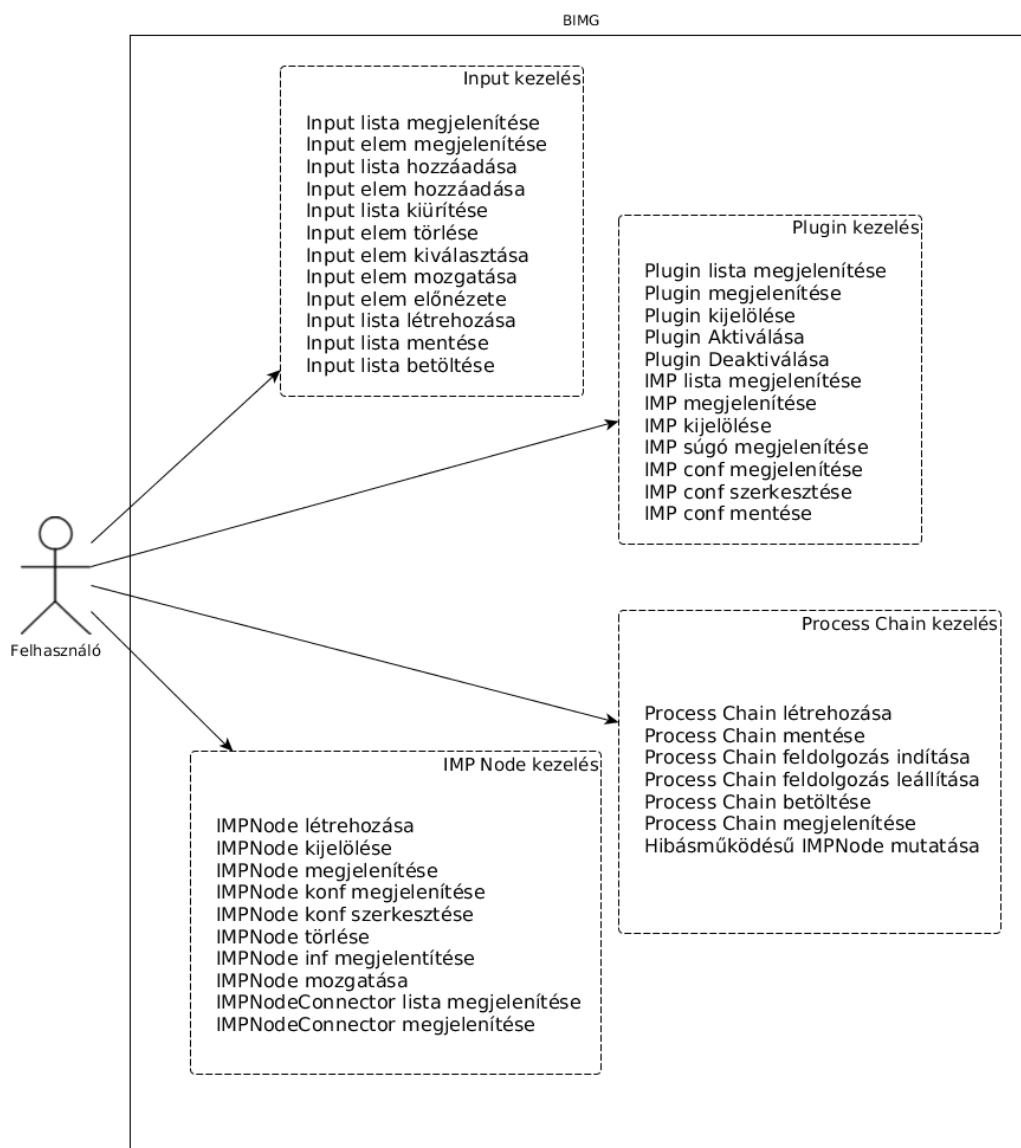
- Input kezelés: itt történik az input hozzáadása, törlése, szerkesztése, az input elemek sorrendjének megváltoztatása, továbbá a bemeneti elemek megjelenítése. _____
- Plugin kezelés: itt történik a pluginek a rendszerbe töltése, ki/be kapcsolása, és tartalmuk böngészése is. _____
- Process Chain kezelés: a feldolgozás szíve, ez a modul veszi át az input listát, és végzi a feldolgozást
- IMP Node kezelés: Itt történik a Process Chain elemeinek a definiálása és az elemek közötti relációknak a beállítása. _____

Törölni!

Törölni!

Törölni!

3. RENDSZERTERVEK



2. ábra. A BIMG sematikus usecase diagrammja

Ezek a logikai csoportok reprezentálnak egyben egy modult is a rendszerben. A teljes usecase kifejtve a mellékletben olvasható.

USECASE

3.3.2. Nem funkcionális követelmények

3.4. Választott technológiák

A megfelelő technológiák kiválasztása során több tényező is befolyásolta a döntésemet. Ezekekből két nagy csoportot írtam fel: követelményrendszerből illetve szubjektív nézőpontból kiemelt fontosságú tényezők. A követelményrendszer-

3. RENDSZERTERVEK

ből adódóakat a korábbi két alpontban bőségesen részleteztem, ezért a következőkben csak a szubjektív pontokat vázolnám fel.

- Egyszerű és gyors, minőségi fejlesztés
- Könnyű dokumentálhatóság
- Legyen korábbi munkáimból rutinom az adott technológiák alkalmazásában
- Képfeldolgozási függvénykönyvtárakkal legyenek jól ellátottak a kiválasztott technológiák (nem szeretném újra feltalálni a kereket)

A fenti szempontokból a következő fegyvertár került összeállításra:

A program alap szerkezete Qt-val, a képfeldolgozásért felelős komponensek OpenCv-vel kerültek megvalósításra. A verziókövetést Git-el végeztem, a dokumentáció és dolgozat elkészítéséhez pedig Latex-et, Gummi-t, és Yed-et használtam.

3.4.1. Qt

Egyike a legmeghatározóbb multiplatform c++-ra épülő alkalmazás keretrendszerek. [13] Korábban már több másik projektben is sikeresen dolgoztam vele. A részletes dokumentáció és aktív felhasználói/fejlesztői bázis sokat segített a fejlesztésben. [14] [15][16] Licencelése kedvező, elérhető OpenSource és Enterprise verziója is. Olyan nagy cégek is használják mint a BlackBerry, Michelin vagy a Panasonic.[17] Jelen dolgozat tárgyát képező alkalmazás az 5.1.1-es (GCC 4.6.3 32bit) verzióval készült.

3.4.2. OpenCV

Open Source Computer Vision Library, nyíltforrású képfeldolgozást és gépi tanulást megvalósító függvénykönyvtár. Natív c++-ban implementált és erősen támaszkodik az stl tárolókra. Egy vékony wrapper elkészítésével könnyen összekapcsolható Qt-val. Funkcionalitásával széleskörű feladatok megoldására kiválóan alkalmas és dokumentációja is megfelelő. Jelen program a 2.4.6.1-es verzióval került implementálásra.

4. IRODALOMJEGYZÉK

3.5. Architektúra

3.6. Felület terv

4. Irodalomjegyzék

- [1] Craig Larman (2004). Applying UML and Patterns, Prentice Hall, 3 edition 6.7 (66)
- [2] Jack J. Dongarra (1995). Numerical Linear Algebra on High-Performance Computers (3)
- [3] http://www.valvesoftware.com/publications/2006/SIGGRAPH06_Course_ShadingInValvesSourceEngine_Slides.pdf Valve, Jason Mitchell (2007), Shading in Valve's Source Engine (37)
- [4] <http://imagej.nih.gov/ij/> ImageJ - Image Process and Analysis in Java
- [5] Tony J. Collinsm, ImageJ for microscopy BioTechniques 43:S25-S30 (July 2007)
- [6] <http://www.highmotionsoftware.com/products/imbatch>
ImBatch - Batch Image Processing Software
- [7] <http://www.originlab.com/index.aspx?go=Products/Origin/DataAnalysis/ImageProcessing>
OriginLab - Image Processing
- [8] <http://www.originlab.com/index.aspx?go=COMPANY/AboutUs>
OriginLab - About Us
- [9] http://www.originlab.hu/Originv9_USD_NEW_20121022_WEB.pdf
OriginLab - Licences
- [10] <http://www.ibm.com/developerworks/rational/library/4706.html>
IBM - Capturing Architectural Requirements

5. MELLÉKLETEK

- [11] <http://zone.ni.com/reference/en-XX/help/371361J-01/lvconcepts/blockdiagram/>
NI - LabVIEW 2012 Help - Block Diagram
- [12] <https://docs.unrealengine.com/latest/INT/Engine/Blueprints/Editor/index.html>
UDK4 - Blueprint Editor Reference
- [13] <http://qt.digia.com/About-Us/> QT - About
- [14] <http://qt-project.org/doc/> QT - Doc
- [15] <http://qt-project.org/forums> QT - Forums
- [16] <http://lists.qt-project.org/mailman/listinfo> QT - MailingLists
- [17] <http://qt.digia.com/Qt-in-Use/> QT Digia - In Use
- [18] <http://opencv.org/about.html> OpenCV - About
- [19] <http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf>
Guide to the Software Engineering Body of Knowledge, Chapter 2 -
SOFTWARE REQUIREMENTS
- [20] <http://www.ibm.com/developerworks/rational/library/4706.html#N10098>
IBM - Capturing Architectural Requirements, Functional Requirements

5. Mellékletek

5.1. Hasonló célú rendszerek összehasonlítása táblázat

	ImageJ	ImBatch	OriginLab
Platform	Multi (Java)	Win (C#)	Win (C/C++)
Licence	Public Domain	@TODO	@TODO
Cél csoport	professzionális (orvosi)	félprofesszionális (általános)	professzionális (tudományos, ipari)
Támogató	National Institutes of Health	High Motion Software	OriginLab
Felhasználói közösség	wiki, leírások, fejlesztői dokumtáció, levlista, fórum	gyik, leírások, oktató videók	gyik, wiki, leírások, fejlesztői dokumentáció, fórum
Plugin rendszer	igen	igen	igen
Kötegetelt feldolgozás	igen (Z-T funkciók)	igen (akár helyérzékeny menü)	igen
Automatizálás	makrók	részben	originC
Fejlesztői eszközök	igen	igen	igen
Bemeneti formátumok	széleskörű (orvosi irány)	széleskörű (általános irány)	széleskörű (ipari irány)
Megjelenítés, GUI	komplex	egyszerű letisztult	komplex
Képjavító eljárások	igen	igen	igen
Geometriai műveletek	igen	igen	igen
Analizálás	igen	nem	igen
Szerkesztési műveletek	igen	igen	igen
Szinterek közötti konverzió	igen	igen	igen

1.2. táblázat. Hasonló célú rendszerek összehasonlítása

5. MELLÉKLETEK

5.2. Teljes usecase

5.3. CD melléklet

A szakdolgozat CD mellékletének könyvtárszerkezete:

/VargaMarcell-DVLKHU-szakdolgozat.pdf

/szakdolgozat-forraskod

/diagramok

/szakdolgozat.tex

/rendszer-forraskod

/src

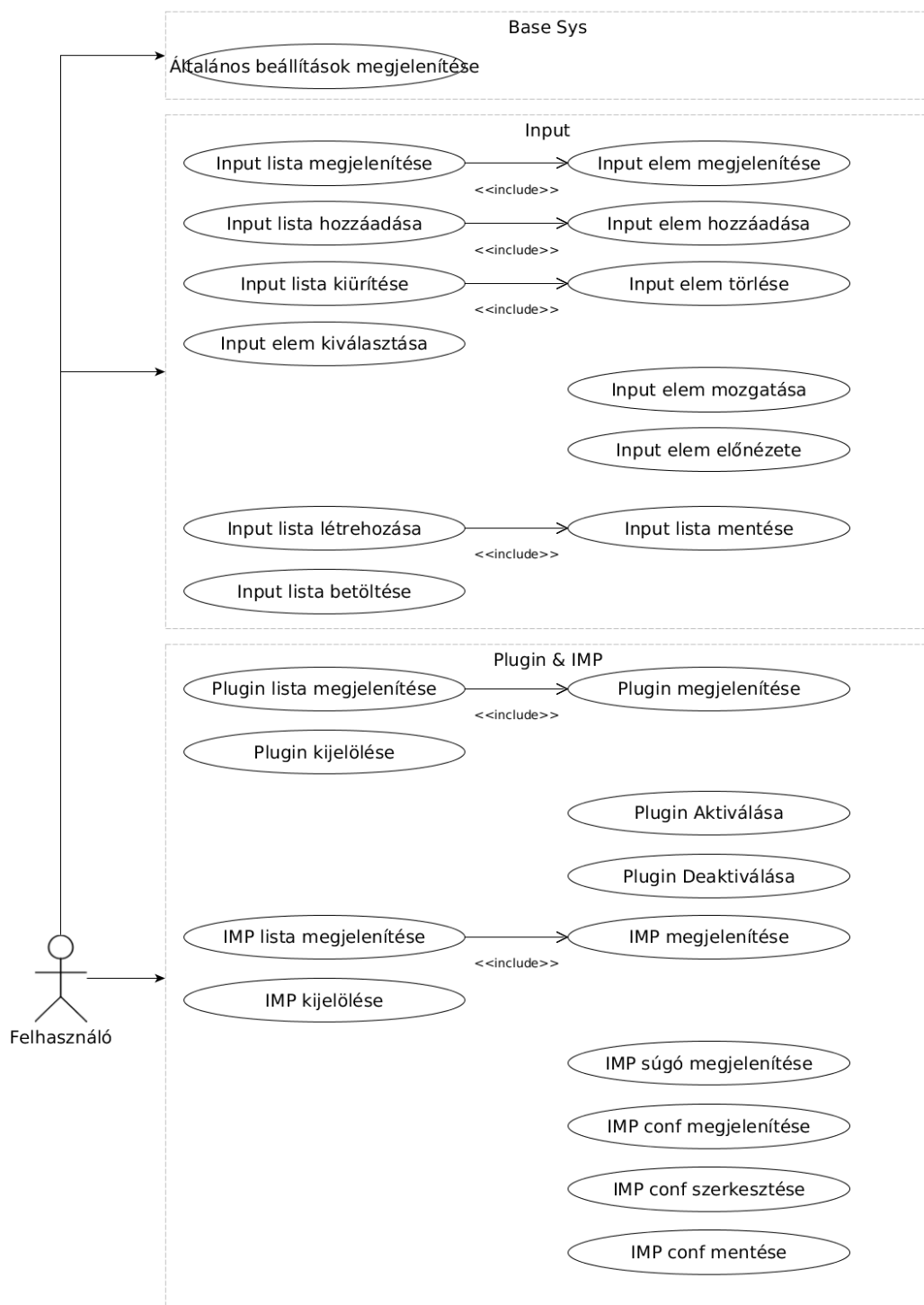
/Szakdolgozat

/internetes-hivatkozasok

/1_osszehasonlitas

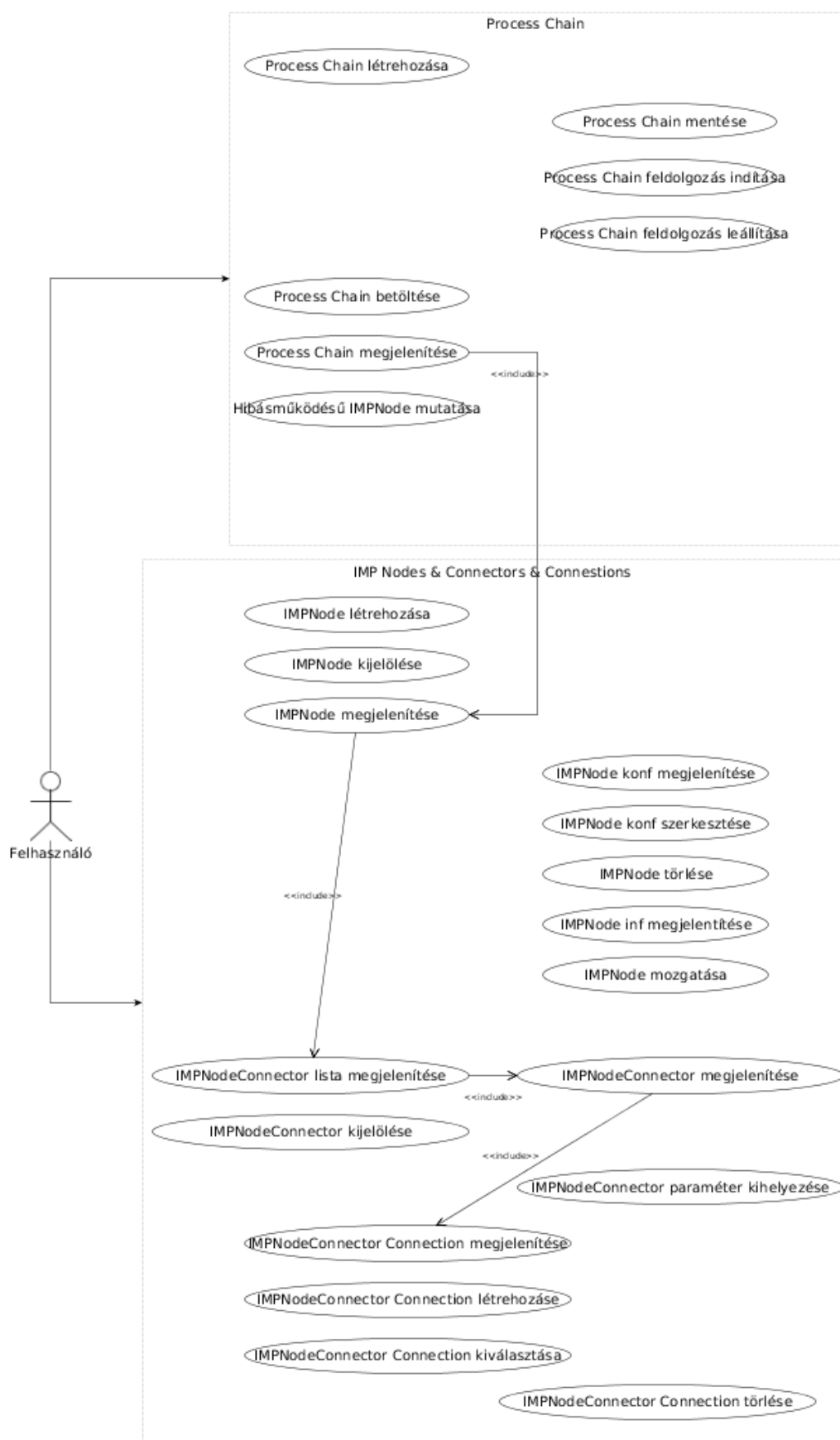
/2_kovetelmenyanalisis

5. MELLÉKLETEK



3. ábra. A BIMG teljes usecase diagrammja (1. rész)

5. MELLÉKLETEK



4. ábra. A BIMG teljes usecase diagrammja (2. rész)